Universal approximation and error bounds for Fourier Neural Operators

Nicolas Nytko, Sean Farhat, Nathanael Assefa

April 25, 2023

Overview

1. Definition of Fourier Neural Operators
2. Universal Approximation Theorem
3. Comparison to DeepOnets
4. Stationary Darcy Flow
5. Incompressible Navier-Stokes

Problem Setting

- Operators are *maps between function spaces*.
- Say we are working in some subspace $D \subset R^d$.
- Let $\mathcal{A}(D; \mathbb{R}^{d_a})$ and $\mathcal{U}(D; \mathbb{R}^{d_u})$ be spaces of functions from D to $\mathbb{R}^{d_a}$ and $\mathbb{R}^{d_u}$, respectively.
- Want to learn some $\mathcal{G}$ that maps from $\mathcal{A}$ to $\mathcal{U}$.
  - Why?
  - Example: PDEs. Map from a right-hand-side to a solution function.

Neural Operator

For some $a \in \mathcal{A}$, we can define a Neural Operator like

$$\mathcal{N}(a) = \mathcal{Q} \circ \mathcal{L}_L \circ \mathcal{L}_{L-1} \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{R}(a). \tag{1}$$

Where

$$\mathcal{R} : \mathcal{A} \to \mathcal{U}(D; \mathbb{R}^{d_v}), \tag{2}$$
$$\mathcal{Q} : \mathcal{U}(D; \mathbb{R}^{d_v}) \to \mathcal{U}(D; \mathbb{R}^{d_u}). \tag{3}$$

We use $\mathcal{R}$ to "lift" $a(x)$ to $\mathbb{R}^{d_v}$, then $\mathcal{Q}$ to "project" to $\mathbb{R}^{d_u}$. Assume $d_v \geq d_u$.
These are linear mappings on the function output $a(x)$.

“L” layers in Neural Operator

$$\mathcal{N}(a) = \mathcal{Q} \circ \mathcal{L}_L \circ \mathcal{L}_{L-1} \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{R}(a)$$

The layers $\mathcal{L}_i$ are learned non-linear layers,

$$\mathcal{L}_i(v)(x) = \sigma\Big(W_i v(x) + b_i(x) + \big(\mathcal{K}(a;\theta_i)v\big)(x) \Big). \tag{4}$$

Left terms are affine transform, right term is integral operator

$$\big(\mathcal{K}(a;\theta_i)v\big)(x) = \int_D \kappa_\theta(x, y, a(x), a(y))v(y)\ dy. \tag{5}$$

Convolutional Operator

If instead we parameterize κ_θ like

$$\kappa_\theta(x - y), \tag{6}$$

we can instead write $\mathcal{K}$ as

$$\big(\mathcal{K}(a;\theta_i)v\big)(x) = \int_D \kappa_\theta(x - y)v(y)\ dy. \tag{7}$$

This is exactly a convolution, so we can speed it up via the Fourier Transform.

Fourier Neural Operator

Define $\mathcal{F}$ as the operator mapping from $\mathcal{R}^{d_v}$ to Fourier space, with $\mathcal{F}^{-1}$ denoting its inverse. We get,

$$\big(\mathcal{K}(a;\theta_i)v\big)(x) = \mathcal{F}^{-1}\big(P_\theta(k)\cdot\mathcal{F}(v)(k)\big)(x), \tag{8}$$

so we can compute the kernel $\mathcal{K}$ as a pointwise multiplication with a learnable matrix $P_\theta(k) \in \mathbb{C}^{d_v \times d_v}$.

$$\implies P_\theta(k) = \mathcal{F}(\kappa_\theta)(k).$$

Implementation with Discrete Fourier Transform

Exact Fourier transform is difficult to compute in practice (requires integration), approximate with discrete Fourier transform; denote by Ψ-spectral FNO.

$$\mathcal{N}(a) = \mathcal{Q} \circ \mathcal{I}_N \circ \mathcal{L}_L \circ \mathcal{I}_N \circ \mathcal{L}_{L-1} \circ \mathcal{I}_N \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{I}_N \circ \mathcal{R}(a)$$

The $\mathcal{I}_N$ term is a *pseudo-spectral projection* onto a finite trigonometric polynomial, interpolates exactly at some set of gridpoints.

For numerical implementation, Ψ-spectral FNO can be thought of as the mapping

$$\widehat{\mathcal{N}} : \mathbb{R}^{d_a \times \mathcal{J}_N} \to \mathbb{R}^{d_u \times \mathcal{J}_N},$$

where functions are evaluated on the grid $\{x_j\}_{j \in \mathcal{J}_N}$, $\mathcal{J}_N := \{0, \ldots, 2N\}^d$

Can alternatively denote Ψ-spectral FNO by

$$\widehat{\mathcal{N}}(a) = \widehat{\mathcal{Q}} \circ \widehat{\mathcal{L}}_L \circ \widehat{\mathcal{L}}_{L-1} \circ \cdots \circ \widehat{\mathcal{L}}_1 \circ \widehat{\mathcal{R}}(a),$$

where functions denoted by $\widehat{\cdot}$ are discrete evaluations on some sample points.

Measuring size of $\Psi-$FNO

For a $\Psi-$FNO $\hat{\mathcal{N}}$, grid points $\mathcal{J}_N$, $|\mathcal{J}_N| = (2N+1)^d$, and L layers,

$$\text{size}(\mathcal{N}) = \underbrace{d_u d_v}_{Q} + L\left(\underbrace{d_v^2}_{W_\ell} + \underbrace{d_v|\mathcal{J}_N|}_{b_\ell} + \underbrace{d_v^2|\mathcal{J}_N|}_{P_\theta}\right) + \underbrace{d_a d_v}_{R} \tag{9}$$

Mini-intro to functional analysis

Definition (Sobolev Space)

A vector space of functions, $W^{k,p}(\cdot)$, whose weak partial derivatives up to degree k exist, and are square integrable (L^2).

Special case $k = 2$: $H(\mathbb{T}^d)^p = W(\mathbb{T}^d)^{2,p}$, nice properties with Fourier modes.

$$\|v\|_{H^p}^2 = \frac{(2\pi)^d}{2} \sum_{k \in \mathbb{Z}^d} (1 + |k|^{2p}) \|\hat{v}_k\|^2 < \infty$$

for Fourier coefficients $\hat{v}_k$.

Universal Approximation Theorem

(Theorem 2.5)

Theorem (Universal Approximation)

For $s, s' \geq 0$*, any continuous operator* $\mathcal{G} : H^s(\mathbb{T}^d; \mathbb{R}^{d_a}) \to H^{s'}(\mathbb{T}^d; \mathbb{R}^{d_u})$*, and compact subset of functions* $K \subset H^s(\mathbb{T}^d; \mathbb{R}^{d_a})$*; for any* $\varepsilon > 0$ *there exists some FNO* $\mathcal{N}$ *such that*

$$\sup_{a \in K} \|\mathcal{G}(a) - \mathcal{N}(a)\|_{H^{s'}} \leq \varepsilon.$$

Given a large class of operators, in theory* there is always an FNO that approximates this set to desired accuracy ε

Universal Approximation Sketch

(Theorem 2.5)

Theorem (Universal Approximation)

The main objective is thus to prove Theorem 2.5 for the special case $s' = 0$*; i.e. given a continuous operator* $\mathcal{G} : H^s\left(\mathbb{T}^d\right) \to L^2\left(\mathbb{T}^d\right), K \subset H^s\left(\mathbb{T}^d\right)$ *compact, and* $\epsilon > 0$*, we wish to construct a* $\mathrm{FNO}\, \mathcal{N} : H^s\left(\mathbb{T}^d\right) \to L^2\left(\mathbb{T}^d\right)$*, such that*
$\sup_{a \in K} \|\mathcal{G}(a) - \mathcal{N}(a)\|_{L^2} \leq \epsilon$

To this end, we start by defining the following operator,

$$\mathcal{G}_N : H^s\left(\mathbb{T}^d\right) \to L^2\left(\mathbb{T}^d\right), \quad \mathcal{G}_N(a) := P_N \mathcal{G}\left(P_N a\right)$$

with P_N being the orthogonal Fourier projection operator

Universal Approximation Theorem Proof

(2.5 Proof) We circumvent the later transforms by composing with an additional inverse FNO layer $\widetilde{\mathcal{L}} : L^2 \to H^{s'}$ satisfying the identity $\widetilde{\mathcal{L}}(v) = \widetilde{\mathcal{L}}\left(P_N v\right)$ for all v, and defining a continuous operator $H^{s'} \to H^{s'}$, such that

$$\sup_{v \in K'} \left\| P_N v - \widetilde{\mathcal{L}}(v) \right\|_{H^{s'}} \leq \delta$$

Next, we define a new FNO by the composition $\mathcal{N} := \widetilde{\mathcal{L}} \circ \widetilde{\mathcal{N}} : H^s \to H^{s'}$.$\mathcal{N}$ is a continuous operator $H^s \to H^{s'}$, since it can be

Theorem

Proof [written as the composition]

$$H^s \xrightarrow{\widetilde{\mathcal{N}}} L^2 \xrightarrow{P_N} H^{s'} \xrightarrow{\widetilde{\mathcal{L}}} H^{s'}$$

Given a large class of operators, in theory* there is always an FNO that approximates this set to desired accuracy ε

Remark on super-exponential scaling

(Remark 3.1)
We can construct an FNO approximating an operator $\mathcal{G}$ like

$$\mathcal{N} := \mathcal{N}_{\mathsf{IFT}} \circ \tilde{\mathcal{N}} \circ \mathcal{N}_{\mathsf{FT}},$$

where $\mathcal{N}_{\mathsf{IFT}}, \mathcal{N}_{\mathsf{FT}}$ are the inverse and regular fourier transform approximations, respectively, and $\tilde{\mathcal{N}} : \mathbb{R}^{2\mathcal{K}_N} \to \mathbb{R}^{2\mathcal{K}_N}$ is a neural network operating on Fourier space.

N denotes the number of Fourier coefficients, function of desired accuracy ε. Assuming Lipschitz continuity and that our input function lies in $a \in K \subset H^s$, the width of $\tilde{\mathcal{N}}$ scales asymptotically like

$$\text{width}(\tilde{\mathcal{N}}) \gtrsim \epsilon^{-\epsilon^{-d/s}}$$

Deep Onet Review

A *DeepOnet* is a mapping $\mathcal{O} : C(\mathbb{T}^d; \mathbb{R}^{d_a}) \to C(\mathbb{T}^d; \mathbb{R}^{d_v})$ of the form

$$\mathcal{O}(a)(x) = \sum_{k=1}^{p} \beta_k(a(x_1), \dots, a(x_p))\tau_k(x), \tag{10}$$

where $\beta_i : \mathbb{R}^{m \times d_a} \to \mathbb{R}^{p \times d_u}$ and $\tau_j : \mathbb{R}^d \to \mathbb{R}^p$ are the *branch* and *trunk* networks, respectively.

The function a is evaluated at some discrete *sensor points* $x_1, \dots, x_m$.

A $\Psi-$FNO is a specific formulation of the branch and trunk networks for a DeepOnet.

Theorem (Approximation of $\Psi-$FNO by DeepOnet)

Let $\hat{\mathcal{N}} : L^2_N(\mathbb{T}^d;\mathbb{R}^{d_a}) \to L^2_N(\mathbb{T}^d;\mathbb{R}^{d_u})$ be a $\Psi-$FNO. For any $\varepsilon > 0$ and fixed $B > 0$, there is a DeepOnet with equi-spaced sensor points $x_1, \dots, x_m$; arbitrary branch net β, and trunk net τ that by construction approximates some orthonormal trigonometric basis. We can bound the error between them like

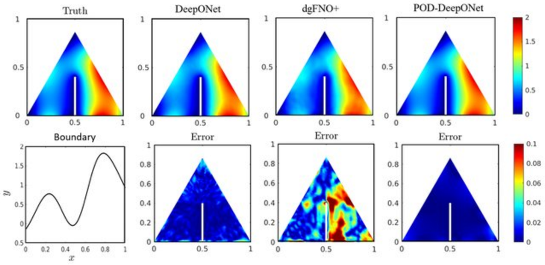
$$\sup_{\|a\|_{L^\infty} \leq B} \sup_{y \in \mathbb{T}^d} \left| \hat{\mathcal{N}}(a)(y) - \sum_{k=1}^{p} \beta_k(a)\tau_k(y) \right| \leq \varepsilon$$

The width and depth of the branch-net is equal to that of $\hat{\mathcal{N}}$. However, the size of $\hat{\mathcal{N}}$ is much smaller than that of the branch and trunk networks (of the constructed emulator).

Stationary Darcy Flow

$$-\nabla \cdot (a\nabla u) = f. \tag{11}$$

On a periodic domain $\mathbb{T}^d$, and assuming that our solution has zero mean.



[1]

(Figure is not the same problem)

[1] Physics-Informed Deep Neural Operator Networks, Goswami et al.

$$-\nabla \cdot (a\nabla u) = f. \tag{11}$$

We will impose that $\int_{\mathbb{T}^d} f \, dx = \int_{\mathbb{T}^d} u \, dx = 0$, such that the RHS and solution have zero mean. Let $a = 1 + \tilde{a}$, for $\tilde{a} \in H^s(\mathbb{T}^d)$, $s > d/2$. Clasically, we solve the Fourier-Galerkin approximation,

$$-\dot{P}_N \nabla \cdot ((1 + \dot{P}_N \mathcal{I}_{2N} \tilde{a}) \nabla u_N) = \dot{P}_N \mathcal{I}_{2N} f, \tag{12}$$

on an equispaced, regular grid $\{x_j\}_{j \in \mathcal{J}_{2N}}$.

Want to use $\Psi-$FNO to approximate

$$\mathcal{G} : L^\infty(\mathbb{T}^d) \to H^1(\mathbb{T}^d) \tag{13}$$

$$\mathcal{G} : a \mapsto u \tag{14}$$

Approximation by $\Psi-$FNO

Theorem (Existence of Discrete FNO)

Assuming activation is smoothly differentiable three times, there exists $C > 0$, $k \in \mathbb{N}$, *and an* $\Psi-$*FNO* $\hat{\mathcal{N}} : H^s(\mathbb{T}^d) \to H^1(\mathbb{T}^d)$ *such that*

$$\textit{depth}(\hat{\mathcal{N}}) \leq C \log(N), \quad \textit{lift}(\hat{\mathcal{N}}) \leq C, \quad \textit{width}(\hat{\mathcal{N}}) \leq CN^d$$

such that

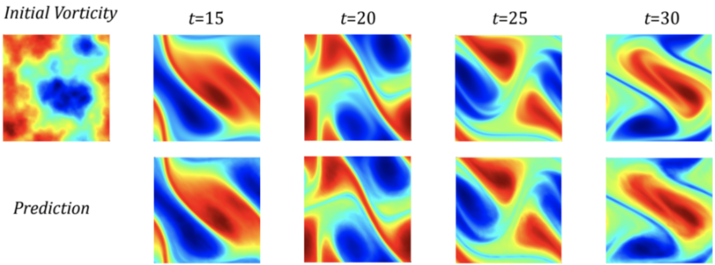
$$\sup_{a \in H^s} \|\mathcal{G}(a) - \hat{\mathcal{N}}(a)\|_{H^1(\mathbb{T}^d)} \leq CN^{-k}$$

Incompressible Navier-Stokes

$$\frac{\partial u}{\partial t} + u \cdot \nabla u + \nabla p - \nu \nabla^2 u = 0 \tag{15}$$

$$\nabla \cdot u = 0 \tag{16}$$

$$u_0 = u(t = 0). \tag{17}$$



[2]

[2] Fourier Neural Operator for Parametric Differential Equations, Li et al.

Leray Projection

Definition (Leray Projector)

Let $\mathbb{P} : L^2(\mathbb{T}^d;\mathbb{R}^d) \to \dot{L}^2(\mathbb{T}^d;\text{div})$ be the L^2-orthogonal projector onto the set of functions with zero mean and no divergence. We can explicitly write the projection operator like

$$\mathbb{P}\left(\sum_{k\in\mathbb{Z}^d} \hat{u}_k e^{i\langle k,x\rangle}\right) = \sum_{k\in\mathbb{Z}^d\setminus\{0\}} \left(1 - \frac{k\otimes k}{\|k\|^2}\right)\hat{u}_k e^{i\langle k,x\rangle}. \tag{18}$$

We can thus rewrite the Navier-Stokes equations as

$$\frac{\partial u}{\partial t} = -\mathbb{P}(u\cdot\nabla u) + \nabla^2 u \tag{19}$$

Discretization in Time (Classical Formulation)

We use an implicit Euler time discretization (with $\Delta t > 0$), to get the recurrence

$$\frac{u_N^{n+1} - u_N^n}{\Delta t} + \mathbb{P}_N(u_N^n \cdot \nabla u_N^{n+1}) = \nu \nabla^2 u_N^{n+1}. \tag{20}$$

With the initial condition $u_N^0 = \mathcal{I}_N u(t = 0)$ projected onto the Fourier representation.

From CFD, we require that the CFL condition,

$$(\Delta t) \| u_N^n \|_{L^\infty} N \leq \frac{1}{2}, \tag{21}$$

is satisfied for convergence.

Fixed-Point Iteration

The update

$$\frac{u_N^{n+1} - u_N^n}{\Delta t} + \mathbb{P}_N(u_N^n \cdot \nabla u_N^{n+1}) = \nu \nabla^2 u_N^{n+1} \tag{20}$$

is nonlinear, so to solve we re-cast as a fixed point iteration

$$w_N \mapsto (1 - \nu(\Delta t)\nabla^2)^{-1} u_N^n - (\Delta t)(1 - \nu(\Delta t)\nabla^2)^{-1}\mathbb{P}_N(u_N^n \cdot \nabla w_N), \tag{22}$$

repeatedly compute/update to get next timestep.

Approximation by $\Psi-$FNO

Previous computations are *annoying*, can we approximate by FNO?

Theorem (Existence of Discrete FNO)

Assuming activation is smoothly differentiable three times, there exists $C > 0$ and an $\Psi-$FNO $\hat{\mathcal{N}}$ such that

$$\textit{depth}(\hat{\mathcal{N}}), \textit{lift}(\hat{\mathcal{N}}) \leq C, \quad \textit{width}(\hat{\mathcal{N}}) \leq CN^d$$

such that

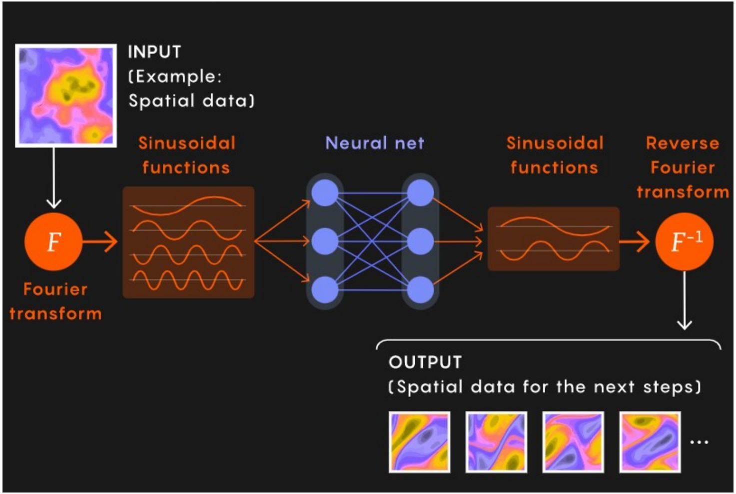
$$\|\mathbb{P}_N(u_N - \nabla w_N) - \hat{\mathcal{N}}(u_N, w_N)\|_{L^2_N} \leq \varepsilon.$$

Nonlinearities can be approximated by pseudospectral FNOs!

Bibliography

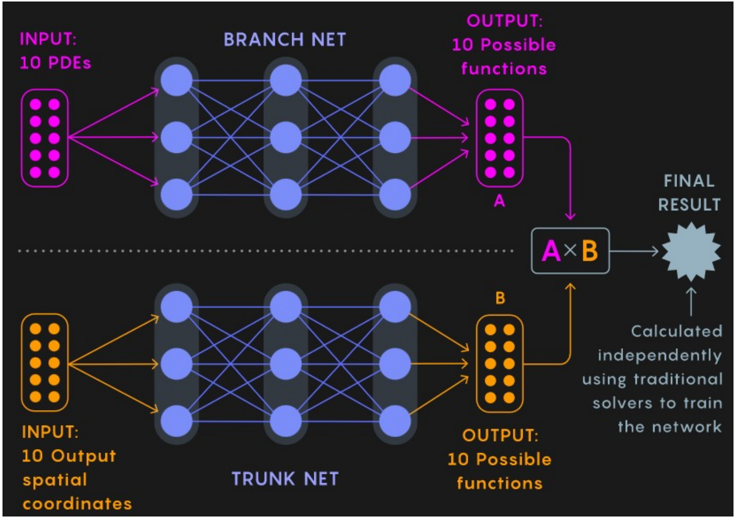
- **On Universal Approximation and Error Bounds for Fourier Neural Operators**, Nikola Kovachki, Samuel Lanthaler, Siddhartha Mishra
- **Fourier Neural Operator for Parametric Partial Differential Equations**, Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar
- **Physics-Informed Deep Neural Operator Networks**, Somdatta Goswami, Aniruddha Bora, Yue Yu, George Em Karniadakis

Fourier Neural Operator Architecture



https://www.quantamagazine.org/latest-neural-nets-solve-worlds-hardest-equations-faster-than-ever-before-20210419/

DeepONet Architecture



https://www.quantamagazine.org/latest-neural-nets-solve-worlds-hardest-equations-faster-than-ever-before-20210419/